TARTAID (Version 2002-1)

by
Dermott E. Cullen
University of California
Lawrence Livermore National Laboratory
L-159
P.O. Box 808
Livermore, CA 94550

Tele: 925-423-7359
E.Mail: cullen1@llnl.gov
Website: http://www.llnl.gov/cullen1

Introduction

This code is designed to allow you to quickly prepare TART input decks starting from scratch. Its most effective use is to define simple objects, in great detail using many spatial zones. You can then combine these simple objects into more complicated geometry.

WARNING – In the current distribution of **TARTAID**, the TART deck created is named **TART.IN**; this is in contrast to the earlier version of **TARTAID**, where the deck created was named **TART.OUT**. With the earlier convention is was too easy to forget to rename the deck for input to TART, with the result that after spending the time to create an input deck, it was often overwritten as soon as TART started running. With the present convention the **TART.IN** file created by **TARTAID** already has the TART default input file name.

For compatibility with earlier versions of TART and all existing older TART input decks, TART MUST use the same default parameters that it has always used. Today these default parameters are not necessarily the best choices to use, but to maintain compatibility that's what we are stuck with. **TARTAID** has no such limitations, and its default values are designed to be the best choices today.

In summary, **TARTAID** helps you with two of the most difficult problems facing anyone who is preparing TART input,

1) Accurately defining geometry.
2) Selecting the Best options to meet today's needs.

TARTAID is an interactive graphics code that uses a screen layout that is very similar to **TARTCHEK**. So if you are familiar with **TARTCHEK**, you will find it very easy to learn TARTAID, and to use it to good advantage.

Overview

Creating a TART Input Deck requires the following steps,
1) Define General Running Conditions.
2) Define ALL Materials for your Problem.
3) Define your Spatial Zones.
4) Assign a Material to EACH Zone.
5) Define your Source.
6) Define Weights.
7) Define Tally and Output Options.

The presently distributed version of **TARTAID** ONLY does steps 1) through 4). Currently you have to do steps 5) through 7) manually. Fortunately, for most TART problems steps 1) through 4) involves the bulk of the work required to prepare TART input, and is by far the most error prone part of input preparation. So you can use **TARTAID** to quickly get you through these steps, and it is then usually much easier to deal with the remaining input preparation, see, the below example.

TARTAID Creates a Complete TART Input Deck

Even though TARTAID only allows you to define everything required for steps 1) through 4), it uses default parameters to define everything required to produce a complete TART input file, and at the end of each run it outputs the file as **TART.IN**. You can then edit **TART.IN** to define parameters for steps 5) through 7). In most cases this only involves defining the source specific to your problem, see, the below example.

Error Checking and Correction

There is a minimum of error checking and correction in the current code. Basically if you input anything that the codes recognizes as illegal, your input is ignored. For example, the density of all materials MUST be positive, and the radii of ALL cylinders and spheres MUST be positive - if you input contradicts this requirement it will be ignored, the you will be given another chance to enter your input.

However, if you input anything that the code finds acceptable, there is no way for you to later change it. For example if by mistake you input the density of water as 10.0 grams/cc, instead of 1.0, you are stuck with it. From a practical viewpoint this isn't as hard to deal with as you might think. Just get used to not worrying about such problems, since you will be able to easily correct it later by editing the output file **TART.IN**. Once you get used to using **TARTAID** you will find that you can zip through it and create an entire TART deck in minutes. So if you make a major input error it is usually easiest to abort the code and start over. Below I'll discuss what's a major input error.

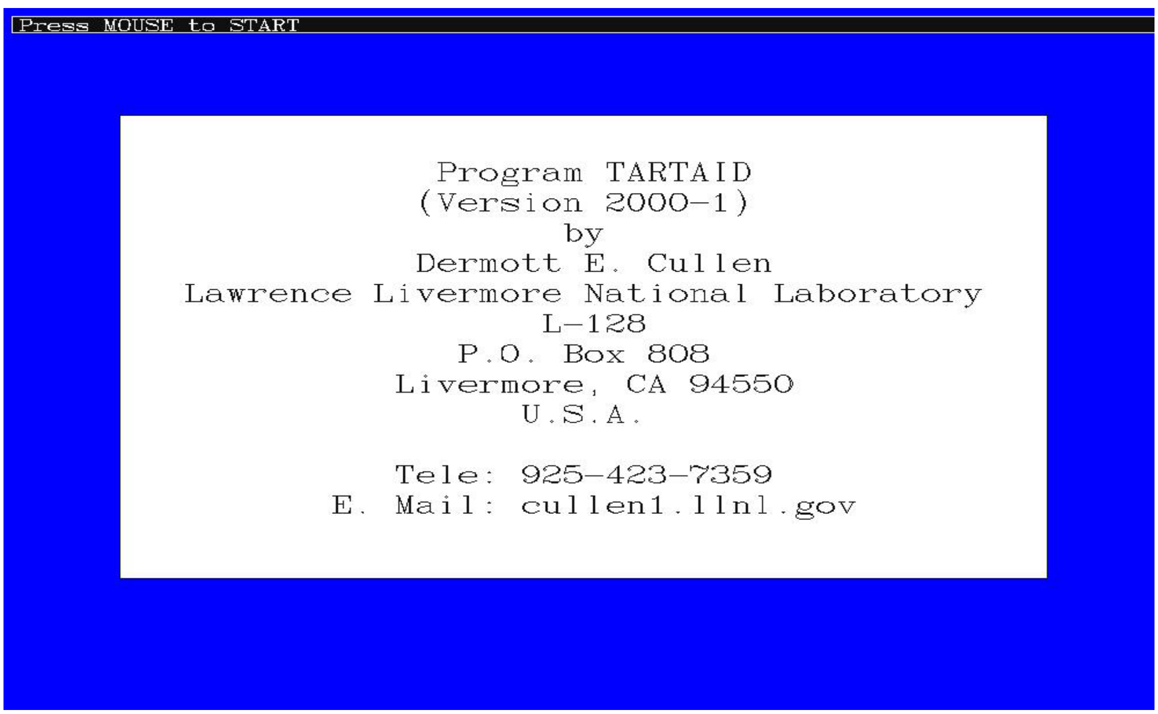
The Quickest Way to Learn to Use TARTAID, is to Use IT

So that's what we are going to do. The only thing that I will assume is that you are generally familiar with TART input parameters, so that as we go through each step using TARTAID, you can generally relate what we are inputting on the screen to how it will appear in a typical TART input deck.

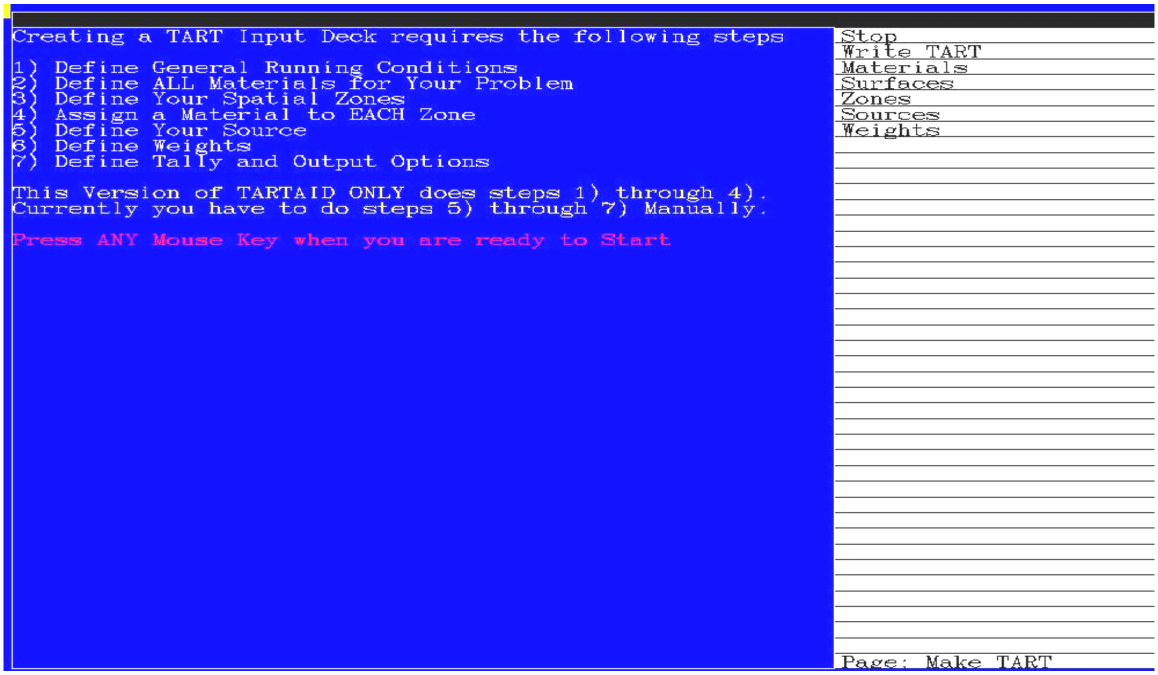
WARNING - the biggest problem that I encounter when trying to teach users to use any interactive graphics code is to teach them to **RELAX and SLOW DOWN, THE COMPUTER WILL WAIT.** When users are confronted with a computer screen that is asking them for input they often make the mistake of feeling that the computer is pressuring them to immediately respond. It's NOT!!! Think about it: the dumb computer isn't pressuring you - it will sit there forever and wait for your input - you are in control. So take your time and slow things down to a pace that is comfortable for you.

WARNING - An IMPORTANT point to remember is that every time you run **TARTAID** it will output results into a file named **TART.IN**. So if you don't want to lose your output from one run, before you run **TARTAID** again, remember to rename **TART.IN**.

Now you start TARTAID running, and I'll talk you through creating your first TART input deck. After that I think you will find it to be easy to use the code on your own.



Screen 1: When you first start TARTAID running the first screen will identify the version of TARTAID that you are running - currently Version 2002-1. Press ANY mouse key to proceed to the next screen. Note, if you report an error or problem to me it is important that I know what version of the code you are using.

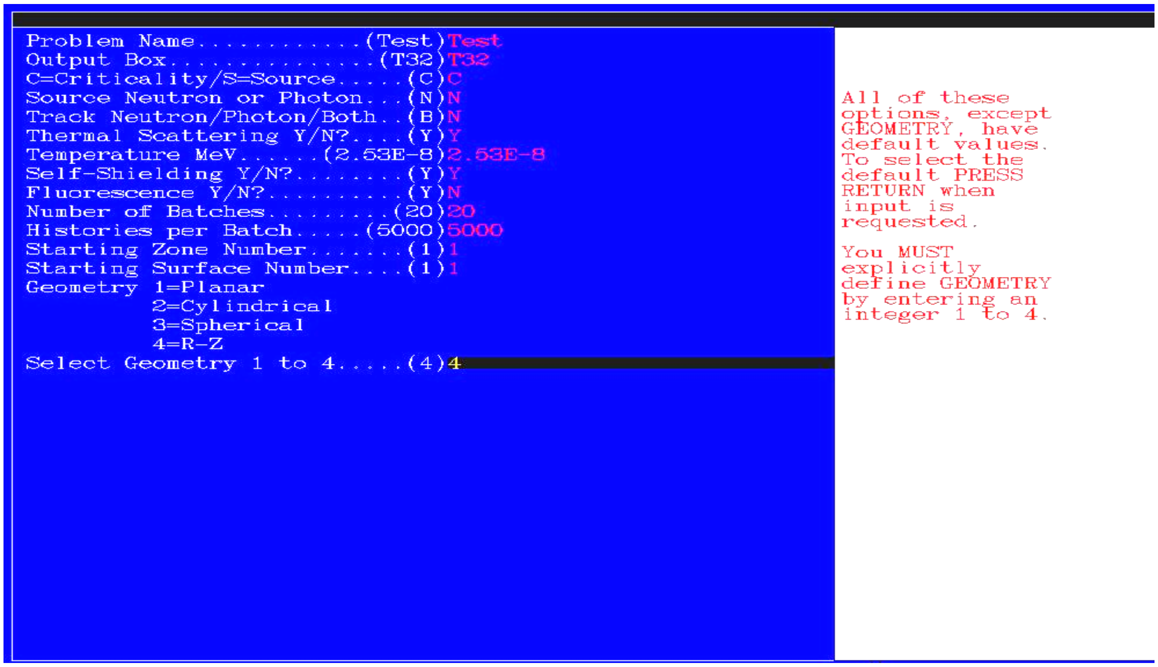


Screen 2: The next screen merely repeats what I said above, namely,

Creating a TART Input Deck requires the following steps,
1) Define General Running Conditions.
2) Define ALL Materials for your Problem.
3) Define your Spatial Zones.
4) Assign a Material to EACH Zone.
5) Define your Source.
6) Define Weights.
7) Define Tally and Output Options.

This version of TARTAID ONLY does steps 1) through 4).
Currently you have to do steps 5) through 7) Manually.

Press ANY Mouse Key when you are ready to Start.



Screen 3: On the next screen you MUST define General Running Conditions. All of these conditions have default values, EXCEPT for the last input parameter, defining your type of geometry. I've tried to keep this input as simple and as short as possible. When you are given choices usually responses are only one letter, e.g., N for Neutron, P for Photon, Y for Yes, N for No, etc.. In ALL cases these responses are case insensitive, e.g., you can use upper case N or low case n, for Neutron. If you make a mistake and the code doesn't accept your input, it will remove your input, display a black area and wait for you to input again. If you make a mistake, but the code accepts it, don't worry about it; you can fix it later in the file **TART.IN**.

Here's what you will see on the screen - only you will see it one line at a time, and you are expected to input something before preceding to the next line. Here each line prompts you to define a TART general running condition, and tells you what the default will be if you press the ENTER key without first inputting ANY characters.

Problem Name…………………………(Test)
Output Box……………………………..(T32)
C=Criticality/S=Source…………………(C)
Source Neutron or Photon……………….(N)
Track Neutron/Photon/Both……………..(B)
Thermal Scattering Y/N?………………..(Y)
Temperature MeV………………...(2.53E-8)
Self-Shielding Y/N?……………………..(Y)
Fluorescence Y/N?………………………(Y)
Number of Batches………………………(20)
Histories per Batch……………………(5000)
Starting Zone Number…………………….(1)
Starting Surface Number………………….(1)

Geometry 1 = Planar
2 = Cylindrical
3 = Spherical
4 = R-Z
Select Geometry 1 to 4…………………..(4)

In most cases it is recommended that you use the default values, i.e., just press ENTER to proceed to the next line. Remember if you change your mind later it is trivial to change any of these parameters in the file **TART.IN**. So don't worry too much about what you enter here. The only things to consider at,

1) **Do you want to run a criticality or source problem?** If you say criticality [enter c or C] your TART input deck will include a critcalc input line. If you say source [enter s or S] your TART input deck will not include a critcalc input line. If you say criticality, you can then only have a neutron source, and can only track neutrons - no photons. In this case you will not be given a choice of some of the following input parameters; all options to do with photons will be defined for you.

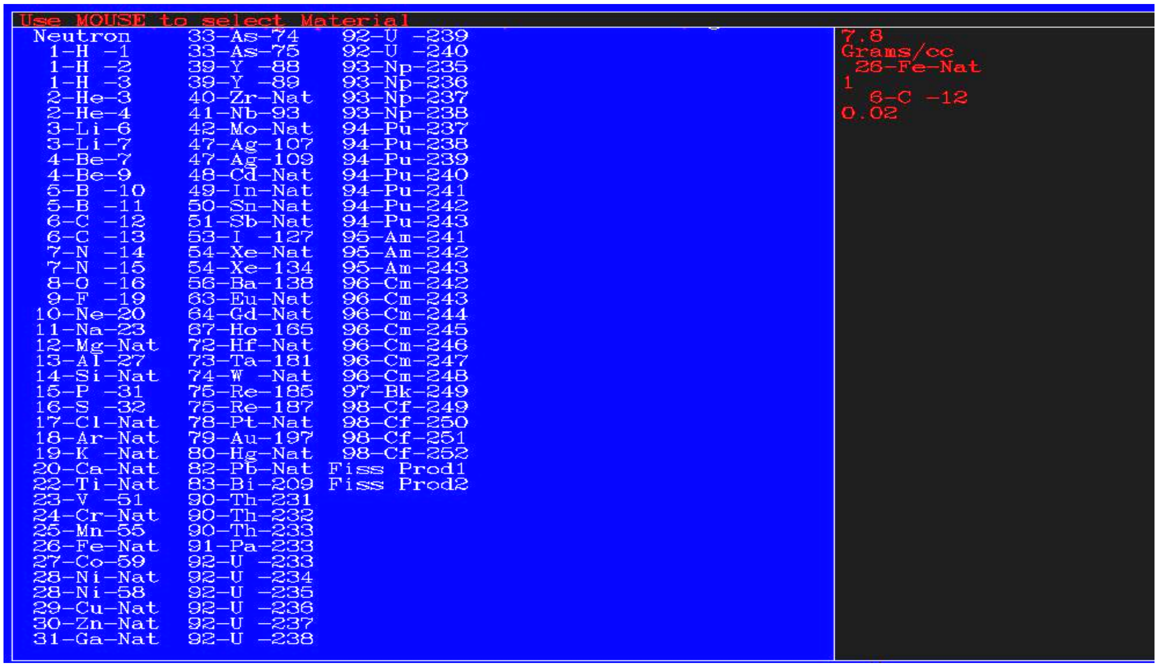
2) **For source problems do you want a neutron or photon source?** You are not allowed to specify both.

3) **For source problems what do you want to track?** This only applies if you want to run a neutron source problem. This is the only case in which you can decide to track only neutrons or neutrons and photons (photons produced by neutron interactions).

4) **Starting Zone and Surface numbers?** For now use the default values. Later we will return to this input when we get to a more complicated problem.

5) **Geometry?** This current version only treats fairly simply 1-D or 2-D geometry.

For now let's skip through all of this input and use the default value for everything until we come to geometry. To do this just press your ENTER key repeatedly until you come to Geometry. At Geometry press your ENTER key again. You will see a black area displayed, indicating that your input was illegal, and giving you another chance to enter a legal parameter. In this case you MUST enter a number from 1 to 4 followed by your ENTER key. As soon as you enter an acceptable value (1 through 4) for geometry the code proceeds to the next screen. For this example, enter 4 followed by ENTER, to select R-Z geometry.



Screen 4: On the next screen you will define ALL of the materials you want to use in your problem. You will do this in EXACTLY the manner used by TART. For each material you will define,

1) The overall density of the material in grams/cc.
2) Whether the constituent "fraction" is defined in terms of atoms or grams (A or G).
3) Then you select a constituent, i.e., 1-H-1, for hydrogen, or 26-Fe-Nat, for iron. You select a constituent by positioning your mouse and clicking on the material.
4) The then define the constituents "fraction"

You can repeat steps 3) and 4) as many times as necessary to define ALL of the constituents of a material.

Let's start,

1) Click on "Define Material" - the right of the screen will go black.
2) The upper left says "Material Density (grams/cc)" - type 1.0 followed by ENTER to define an overall density of 1.0 grams/cc.
3) The upper left says "Fractions in Grams/cc or Atoms/cc (enter G or A)" - type A followed by ENTER, so that each constituent will be defined by atoms/cc.
4) Next position your mouse over "1-H-1" and click - to select hydrogen.
5) The upper left says "Fraction (Atoms/cc)" - type 2 followed by ENTER.
6) Next position your mouse over "8-O-16" and click - to select oxygen
7) The upper left says "Fraction (Atoms/cc)" - type 1 followed by ENTER.
8) To end the material definition move your mouse into the black area to the right and click.

Congratulations - you just successfully defined water as having a density of 1 gram/cc and composed of 2 atoms of hydrogen for each 1 atom of oxygen. Note that the only absolute value is the overall density in grams/cc. All of the fractions will be normalized for you by TART. For example, in this case you could have said there is 1 atom of hydrogen to 0.5 atoms of oxygen, or any other two numbers that are in a ratio of 2:1.

Note, that as you entered each line of input defining your material, the code interpretation of your input is listed to the right. By the time you have finished defining water this will say,

1.0
Atoms/cc
 1-H-1
2
 8-O-16
1

If you don't remember what you input or you aren't sure how the code interpreted your input, you should occasionally check this list.

WARNING - Let me stop here and mention the most probable problem you will have defining materials - you will forget the order of the input and try to do something with your mouse while the code is expecting you to be typing numbers - this can be very frustrating. You can avoid this by **ALWAYS READ WHAT THE TOP, LEFT HAND LINE OF THE SCREEN SAYS** - this will always tell you what the code expects you to be doing. My experience has been that after selecting "Define Material" and then defining the Density, I forget that I MUST next define whether constituent "fractions" are in Grams or Atoms (G or A). I find myself banging away with my mouse trying to define the first constituent and I am getting frustrated because the stupid code isn't responding. In order to avoid this frustration all I have to do is **READ WHAT THE TOP, LEFT HAND LINE OF THE SCREEN SAYS**, which is this case says,
"Fractions in Grams/cc or Atoms/cc (enter G or A)"

Let's define one more material,
1) Click on "Define Material" - the right of the screen will go black.
2) The upper left says "Material Density (grams/cc)" - type 7.8 followed by ENTER to define an overall density of 7.8 grams/cc.
3) The upper left says "Fractions in Grams/cc or Atoms/cc (enter G or A)" - type G followed by ENTER, so that each constituent will be defined by grams/cc.
4) Next position your mouse over "26-Fe-Nat" and click - to select iron.
5) The upper left says "Fraction (Grams/cc)" - type 1 followed by ENTER.
6) Next position your mouse over "6-C-12" and click - to select carbon
7) The upper left says "Fraction (Grams/cc)" - type 0.02 followed by ENTER.
8) To end the material definition move your mouse into the black area to the right and click.

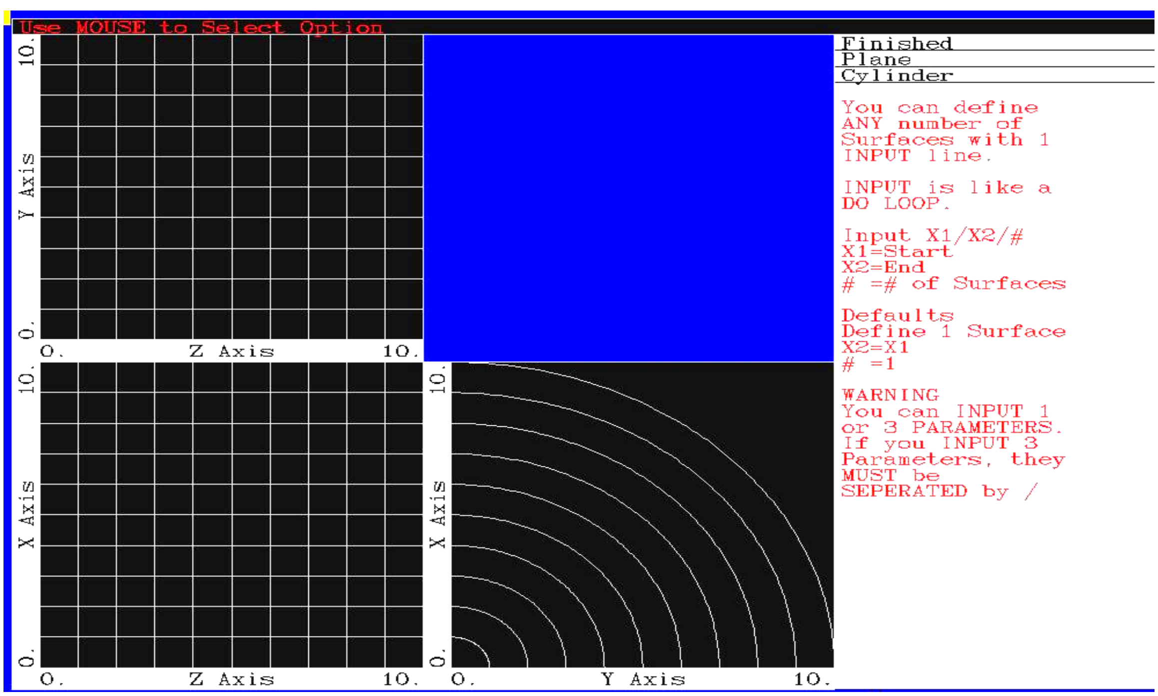
Congratulations - you just successfully defined a simple form of steel as having a density of 7.8 gram/cc and composed of 1 gram of iron for each 0.02 grams of carbon. This is a simple form of steel with 2 % by weight of carbon. Again note that the numbers 1 and 0.02 can be entered as any values as long as the 50:1 ratio is correct.

Note, again that as you entered each line of input defining your material, the code interpretation of your input is listed to the right. By the time you have finished defining steel this will say,

7.8
Grams/cc
 26-Fe-Nat
1
  6-C - 12
0.02

That's as many materials as we need for the steps that follow, so we can now proceed to the next step. However, before we move on, remember here we only defined two materials, and each material had only two constituents. But in a real problem you can define as many materials as you need, and each material can have any number of constituents.

To move on to the next screen select "Finished".



Screen 5: On the next screen you will define ALL of the surfaces of your zones. Since we selected R-Z geometry our surfaces can be planes (Z) and cylinders (R). You can define one simple surface by entering as single number, e.g., 4.3. Or you can define many surfaces at once by entering a lower and upper limit and the number of surfaces you want, e.g., 3.2/9.6/3, says define surfaces at 3.2, 6.4, and 9.6 = 3 surfaces equally spaced between 3.2 and 9.6. If you use ranges there MUST be 3 fields and they MUST be separated by / - if your input isn't in this form it will be ignored and the code will erase your input and wait for your next input.

If you make a mistake and enter something illegal the code will merely ignore your input and cycle back to waiting for correct input. Illegal input includes,
1) Non-positive radius.
2) Non-positive number of surfaces.

The most common error at this point is to think you are defining zones directly, rather than the surfaces that will bound your zones. For example, remember if you want **10** zones equally spaced along the Z axis between 0 and 10 cm, you MUST define **11** planes. However, if you want **10** zones equally spaced radially between 0 and 10 cm, you MUST define **10** cylinders - not **11**, as for planes.

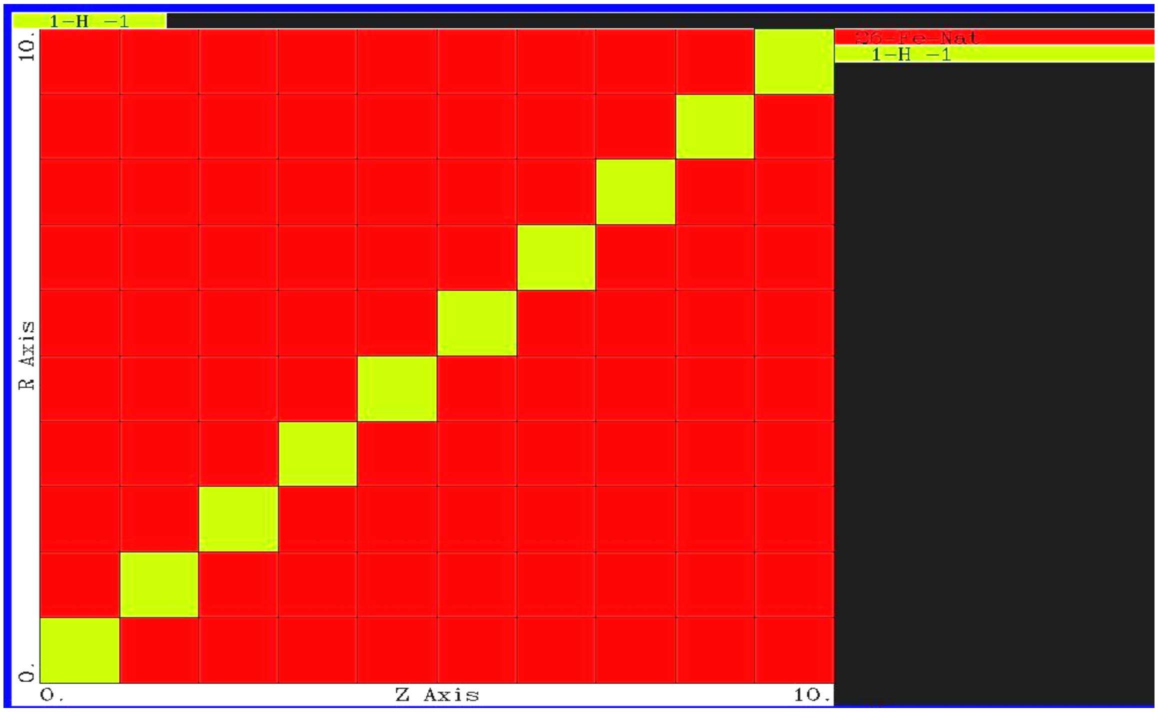
WARNING - Before proceeding let me again stop and point out that this is **THE MOST IMPORTANT STEP IN USING TARTAID TO CREATE TART INPUT**. Defining geometry for any Monte Carlo code that uses 3-D, Combinatorial geometry is VERY DIFFICULT. Therefore if you can avoid difficulties by using **TARTAID** you will have saved yourself a lot of time and effort.

So during this step if you make any mistakes immediately consider aborting and starting over. What's a major mistake? It is a major mistake if you define the wrong number of surfaces. For example, if you want to define 10 cylinders and by mistake you define 100 or 1,000 or even 10,000, IMMEDIATELY stop and start over. The number of zones defined in your output file **TART.IN** is based on how many surfaces you define. So if you get the number of surfaces wrong, your output file will be useless. What's a minor mistake? Surprisingly the wrong positions of surfaces can be a minor mistake. If you don't have too many surfaces, say 10 or so, you can always later edit **TART.IN** to move your surfaces to any position that you want. However, if you have many surfaces, it is better to stop and start over.

Let's see how quickly we can define a 10 by 10 R-Z grid of zones,
1) Click on "Planes"
2) Type 0/10/11 followed by ENTER - 11 planes will be displayed equally spaced along the Z axis between Z = 0 and 10 cm.
3) Click on "Cylinder"
4) Type 1/10/10 followed by ENTER - 10 cylinders will be displayed equally spaced radially.

Congratulation - you just defined a 10 by 10 (100 zones) R-Z grid of zones. Couldn't be easier, right? If you wanted 100 by 100 (10,000 zones) R-Z grid of zones it isn't any more difficult - your input would have been 0/10/101 and 1/10/100 - a piece of cake.

For this example, you have now defined all surfaces, so you can move to the next screen by selecting "Finished".



Screen 6: On the next screen you will assign a material to each zone. To save time the code starts out by assigning the first material that you defined earlier (iron) to each zone –to save yourself time, remember this point, and plan ahead when you use TARTAID for your real problems. Always define the material that is in most of your zones as the first material that you define as **TARTAID** input. **Note that in the above text I described how to define the materials in the order water followed by iron – however the above plot shows that the materials were actually defined in the order iron followed by water – my apologies if this causes any confusion.**

If your problem only included one material, there is nothing else for you to do and you can select "Finished". In this example, earlier we defined two materials (steel and water) so that you can learn how to assign a material to a zone.

To start assigning materials select "Assign Material" and the code will display a list of the materials you defined earlier, where each material is identified by its first defined constituent, and each material is assigned a different color. In this example the list will look like,
26-Fe-Nat (in RED)
1-H -1 (in GREEN)

Note that to start all zones are RED, indicating that iron has been assigned to all zones. To assign water to some of the zone click on the 1-H -1 GREEN area - the selected material will be identified in the upper left. Now to assign water to any zone, position your mouse within the zone and click. You can do this as many times as you like and when you are finished assigning this material move your mouse into the black area to the right and click.

This will bring you back to where you started on this page. You can next again select "Assign Material" to make more changes to the material assigned to each zone, or "Finished".

At this stage if you make a mistake and assign the wrong material to a given zone you can correct yourself by continuing to "Assign Material" until you have everything the way you want it. While you are doing this you can change the material assigned to any given zone any number of times until you are satisfied.

Once you select "Finished" the code will output your **TART.IN** file and it will then terminate. That's it - when you get this to this point you have used **TARTAID** to create a complete TART input deck.

Manually Editing TART.IN

The last step required is to manually edit **TART.IN** to define a source specific to this problem. The example TART input deck that we just created is listed in the appendix. These few simple steps that we followed resulted in a complete TART input deck, in this case 253 lines long.

This includes a default definition of the source as,
1) a neutron source (**sentl 1**).
2) a sphere of radius 1 cm centered on the Z axis and at Z = 5 cm along the Z axis (the default is a source at the center of the defined space) (**source3**)
3) a fission spectrum energy distribution (**sentl 4**)
4) isotropic in direction (**sentl 6 and 7**)

Let's modify **TART.IN** and then demonstrate that the result is a complete TART input deck by using it as input to TART2002. As output by **TARTAID, TART.IN** is already a complete TART input deck, so why do we have to modify it? The only thing that we MUST change is that we accepted the default to run a criticality problem, but there aren't any fissile materials in our problem, so if you try to run it we will get a strange result; try it if you want to see.

Let's change this from a criticality problem to a source problem. To do this we have to make only two changes,
1) First, comment out or delete the "critcalc" line.
2) Change **sentl 2 2000** to **sentl 2 20**

For criticality problems the number of batches (**sentl 2**) is set to a large number to insure convergence is reached before the maximum number of batches (2000) is exceeded; so we assume the code will never really run this number of batches. However, for source problems the code will run exactly the number of batches we tell it to run. If you want to run 2000 batches, that's fine, and no change is required. For our example problem let's run only 20 batches by changing **sentl 2**.

Just so that you get used to doing this, next let's modify the source to be a point source on the Z axis just slightly inside the R-Z geometry, with a 14.1 MeV monoenergetic energy, monodirectional, straight up the Z Axis. As an exercise modify **TART.IN** to include,

* a neutron point source just slightly inside the R-Z geometry
source1 1 0. 0. 1.0e-05
* 14.1 MeV monoenergetic energy
sentl 4 14.1
* monodirectional straight up the Z axis
sentl 6 0.0
sentl 7 1.0

Now you can use this file as input to **TART2002**.

The Real Power of TARTAID

The example that we went through above was relatively simple, but hopefully it got you used to two important concepts,
1) You can use **TARTAID** to quickly produce a TART input deck.
2) You can then modify the deck **TART.IN**.

The next thing to get used to is that the deck **TART.IN** starts with a few run parameters, followed by longer lists of surfaces and zone definitions, followed by a short section defining the source and more run parameters. Most of a typical **TARTAID** output file defines surfaces and zones for a simple 1-D or 2-D geometry. This is the most difficult part of any TART input deck to produce.

The real power of TARTAID is that you can use it to define any number of TART decks, each defining a relatively simple 1-D or 2-D geometry, and you can then edit the decks together to define more complicated geometry. In order to do this let's go back to a few input parameters that we skipped earlier. On **Screen 3** you can define,

Starting Zone Number…………………….(1)
Starting Surface Number………………….(1)

In order to be able to simply combine TARTAID decks all you have to do is,
1) Insure that the various parts of the geometry do not overlap.
2) Insure that the surface and zone numbers of various parts of the geometry do not overlap. This is where the above input can be used.

See if you can put together this problem,
1) A cobalt sphere 0.5 cm in radius centered at Z = 1.0,
2) A lead cylindrical filter 2.0 cm in radius between Z = 4.0 and 4.5 cm,
3) A silicon detector 2.0 cm in radius between Z = 6.0 and 10.0 cm

If you represent each of these three objects as a single spatial zone it should be relatively simple for your to manually create a TART deck. But if you do this you won't learn anything about the spatial distribution of flux or energy deposition in these objects.

So try using **TARTAID** to define a number of TART decks to describe each object in detail. Use 100 radial zones in the cobalt sphere, and 100 by 100 R-Z zones in the lead filter and silicon detector. You can then patch the output results together to define a complete problem. The only zones that you will have to manually define are air zones between the three objects, and vacuum zones surrounding everything; just a hand full of zones. The result will be a problem with over 20,000 zones, and after you run **TART2002** you can immediately run **FLUXEDIT** and then use **TARTCHEK** to see your results overlaid on your geometry. With this approach within a few minutes you will see your results, and you should never have to actually look at the long **TART2002** output report.

Start by using **TARTAID** to define the cobalt sphere. Here you can use the default starting surface and zone numbers. When you finish this **TARTAID** run, first rename **TART.IN** to **DECK1**. Look in this file and find out what is the largest surface and zone number used. Next, define the lead filter. Here use starting surface and zone numbers that are larger than those used to define the cobalt sphere. When you finish this **TARTAID** run, first rename **TART.IN** to **DECK2**. Look in this file and find out what is the largest surface and zone number used. Next, define the silicon detector. Here use starting surface and zone numbers that are larger than those used to define the lead filter. When you finish this **TARTAID** run, first rename **TART.IN** to **DECK3**. Look in this file and find out what is the largest surface and zone number used. The last step is to patch these three decks together and add a few surrounding air zones between objects and vacuum zones surrounding everything, using zone numbers higher than those already used, and you are finished.

Example TARTAID Output File (TART.IN)

```
name Test
box T32 Test
* ==============================================================================
*
* TART Input Deck Generated Using Program TARTAID (Version 2002-1)
*
* ==============================================================================
* ==============================================================================
*
* R-Z Geometry
*
* ==============================================================================
* ==============================================================================
*
* Criticality Problem
* Settle Cycles (15)
* Repetitions (1)
* Standard Deviation (1%)
* Time Step in Shakes (0) Dynamic Problems ONLY
*
* ==============================================================================
critcalc  15 1 10.
* ==============================================================================
*
* Surfaces Definitions
*
* Surface Keyword, #, and Parameters
*
* ==============================================================================
zplane   1 0.
zplane   2 1.
zplane   3 2.
zplane   4 3.
```

```
zplane   5 4.
zplane   6 5.
zplane   7 6.
zplane   8 7.
zplane   9 8.
zplane   10 9.
zplane   11 10.
cylz     12 1.
cylz     13 2.
cylz     14 3.
cylz     15 4.
cylz     16 5.
cylz     17 6.
cylz     18 7.
cylz     19 8.
cylz     20 9.
cylz     21 10.
* ===============================================================================
*
* Material Definitions
*
* matl   # grams/cc Followed by list of ATOM Fractions   and ZA
* matlwp # grams/cc followed by list of WEIGHT Fractions and ZA
*
* ===============================================================================
matl     1 1. 2. 1001 1. 8016
matlwp   2 7.8 1. 26000 0.02 6012
* ===============================================================================
*
* Assign Material to Zones
*
* Material # Followed by Zones Material is Assigned to
*
* ===============================================================================
matz  1 1  thru  44
matz  1 46  thru  55
matz  1 57  thru  66
matz  1 68  thru  77
matz  1 79  thru  99
matz  2 45
matz  2 56
matz  2 67
matz  2 78
matz  2 100
* WARNING - Only legal for exterior, non-re-entrant zones
matz  0 101  thru  103
* ===============================================================================
*
* Neutron Minimum Energy by Zone.
* If Thermal Scattering (sentl 39) is used emin is the
* Temperature in each zone in MeV (2.53d-08)
*
* ===============================================================================
emin  2.53e-8 1  thru  103
* ===============================================================================
*
* Surfaces Bounding Zones
*
* Zone # Followed by Signed Surfaces Bounding Zone
*
* ===============================================================================
jb  1 -1 2 12
jb  2 -2 3 12
jb  3 -3 4 12
jb  4 -4 5 12
jb  5 -5 6 12
jb  6 -6 7 12
jb  7 -7 8 12
jb  8 -8 9 12
jb  9 -9 10 12
jb  10 -10 11 12
jb  11 -1 2 -12 13
jb  12 -2 3 -12 13
jb  13 -3 4 -12 13
jb  14 -4 5 -12 13
jb  15 -5 6 -12 13
jb  16 -6 7 -12 13
jb  17 -7 8 -12 13
jb  18 -8 9 -12 13
jb  19 -9 10 -12 13
jb  20 -10 11 -12 13
```

```
jb  21 -1 2 -13 14
jb  22 -2 3 -13 14
jb  23 -3 4 -13 14
jb  24 -4 5 -13 14
jb  25 -5 6 -13 14
jb  26 -6 7 -13 14
jb  27 -7 8 -13 14
jb  28 -8 9 -13 14
jb  29 -9 10 -13 14
jb  30 -10 11 -13 14
jb  31 -1 2 -14 15
jb  32 -2 3 -14 15
jb  33 -3 4 -14 15
jb  34 -4 5 -14 15
jb  35 -5 6 -14 15
jb  36 -6 7 -14 15
jb  37 -7 8 -14 15
jb  38 -8 9 -14 15
jb  39 -9 10 -14 15
jb  40 -10 11 -14 15
jb  41 -1 2 -15 16
jb  42 -2 3 -15 16
jb  43 -3 4 -15 16
jb  44 -4 5 -15 16
jb  45 -5 6 -15 16
jb  46 -6 7 -15 16
jb  47 -7 8 -15 16
jb  48 -8 9 -15 16
jb  49 -9 10 -15 16
jb  50 -10 11 -15 16
jb  51 -1 2 -16 17
jb  52 -2 3 -16 17
jb  53 -3 4 -16 17
jb  54 -4 5 -16 17
jb  55 -5 6 -16 17
jb  56 -6 7 -16 17
jb  57 -7 8 -16 17
jb  58 -8 9 -16 17
jb  59 -9 10 -16 17
jb  60 -10 11 -16 17
jb  61 -1 2 -17 18
jb  62 -2 3 -17 18
jb  63 -3 4 -17 18
jb  64 -4 5 -17 18
jb  65 -5 6 -17 18
jb  66 -6 7 -17 18
jb  67 -7 8 -17 18
jb  68 -8 9 -17 18
jb  69 -9 10 -17 18
jb  70 -10 11 -17 18
jb  71 -1 2 -18 19
jb  72 -2 3 -18 19
jb  73 -3 4 -18 19
jb  74 -4 5 -18 19
jb  75 -5 6 -18 19
jb  76 -6 7 -18 19
jb  77 -7 8 -18 19
jb  78 -8 9 -18 19
jb  79 -9 10 -18 19
jb  80 -10 11 -18 19
jb  81 -1 2 -19 20
jb  82 -2 3 -19 20
jb  83 -3 4 -19 20
jb  84 -4 5 -19 20
jb  85 -5 6 -19 20
jb  86 -6 7 -19 20
jb  87 -7 8 -19 20
jb  88 -8 9 -19 20
jb  89 -9 10 -19 20
jb  90 -10 11 -19 20
jb  91 -1 2 -20 21
jb  92 -2 3 -20 21
jb  93 -3 4 -20 21
jb  94 -4 5 -20 21
jb  95 -5 6 -20 21
jb  96 -6 7 -20 21
jb  97 -7 8 -20 21
jb  2000 -8 9 -20 21
jb  99 -9 10 -20 21
jb  100 -10 11 -20 21
jb  101 -11
```

```
jb  102 1
jb  103 -21
* ================================================================================
*
* Neutron Source Definitions
*
* ================================================================================
*
* Neutron source position (Point at 0,0,0)
*
* WARNING - Position UNDEFINED - Using DEFAULT
* Spherical Volume nz ri ro z0
* Near the center of the geometry
source3   1 0. 1.0000000149 5.
*
* Neutron source energy (0=Fission spectrum)
*
* WARNING - Energy UNDEFINED - Using DEFAULT
*  4) Neutron Source Energy (0.0)
sentl  4 0.
*
* Neutron source direction (Isotropic, Cosine=-1.0 to +1.0)
*
* WARNING - Angular Distribution UNDEFINED - Using DEFAULT
*   6) Neutron Source Cosine Interval (2.0)
sentl  6 2.
*   7) Neutron Source Cosine Start (-1.0)
sentl  7 -1.
*
* Neutron source initial time (0.0)
*
* ================================================================================
*
* Tally and Output Options
*
* ================================================================================
*   5) Neutron tally type (3)
sentl    5    3
*  33) Photon tally type (3)
sentl   33    3
* ================================================================================
*
* Running Conditions
*
* ================================================================================
*   1) Source and Tracked Particles (0)
sentl    1    1
*   2) Batches of Particles (20)
sentl    2      2000
*   3) Particles per Batch (5000)
sentl    3      1000
*  20) Multiband Self-Shielding (0)
sentl   20    1
*  25) Fluorescence (0)
sentl   25    0
*  39) Thermal Scattering (0)
sentl   39    1
end
```